How Do End-Users Program the Internet of Things?

Original
How Do End-Users Program the Internet of Things? / Corno, Fulvio; De Russis, Luigi; Monge Roffarello, Alberto. - In: BEHAVIOUR & INFORMATION TECHNOLOGY. - ISSN 0144-929X. - STAMPA. - (2022), pp. 1-23. [10.1080/0144929X.2022.2071169]

Publisher:
Taylor & Francis

Published
DOI:10.1080/0144929X.2022.2071169

How Do End-Users Program the Internet of Things?

Fulvio Corno[1], Luigi De Russis[1], and Alberto Monge Roffarello ✉[1]

Politecnico di Torino, Corso Duca degli Abruzzi, 24 Torino, Italy 10129
`{fulvio.corno,luigi.derussis,alberto.monge}@polito.it`

Abstract. Nowadays, end users can exploit end-user development platforms to personalize their Internet of Things ecosystems, typically through trigger-action rules. Unfortunately, within such platforms, users are forced to adopt a unique, *vendor-centric* abstraction: to define triggers and actions, they must specifically refer to every single device or online service needed to execute the intended behaviors. As a consequence, little social and practical benefits of end-user development in this domain have emerged so far. In this paper, we build on the idea that other abstractions besides the *vendor-centric* one are possible, and that the growth of end-user personalization in the Internet of Things may depend on their identification. Specifically, we report on the results of a 1-week-long diary study during which 24 participants were free to collect trigger-action rules arising during their daily activities. First, we demonstrate that users would adopt different abstractions by personalizing devices, information, and people-related behaviors, where the individual is at the center of the interaction. Then, we show that the adopted abstraction may depend on different factors, ranging from the user profile, e.g., their programming experience, to the context in which the personalization is introduced. While users are inclined to personalize physical objects in the home, for example, they often go "beyond devices" in the city, where they are more interested in the underlying information. Finally, we discuss the retrieved results by identifying new design opportunities to improve the relationship between users and the Internet of Things.

Keywords: End-User Development · Internet of Things · Trigger-Action Programming · Abstraction · Diary Study · Context.

Word count of the manuscript $\approx$ 13000

1 Introduction

In the technological era we are living, end users can interact with various connected entities, i.e., smart devices [30] and online services [49], to perform many activities, ranging from receiving personalized notifications to controlling the temperature of their environments. Such an ecosystem, commonly referred to as the Internet of Things (IoT), is influencing our society [28]: through a network of physical objects always connected to the Internet, and a multitude of online services such as social networks and news portals, the IoT may help society in many different ways, e.g., through applications ranging in scope from the individual to entire cities [11]. In this domain, connected entities are often utilized together [53]: by exploiting Trigger-Action Programming (TAP) platforms such as IFTTT[1] and Zapier[2], even users without technical skills can nowadays define IF-THEN rules and personalize the joint behavior of their devices and online services. Rules, in particular, can be instantiated in different contexts, ranging from the home [52,26], e.g., *"if my smart car enters the home geographical area, then set a given temperature on my Nest thermostat"*, to online environments [53], e.g., *"if I am tagged in a Facebook photo, then send me a Telegram message."*

Despite apparent simplicity and flexibility, previous works already demonstrated that these platforms for End-User Development (EUD) present their own set of issues: the expressiveness of the definable IF-THEN rules is often limited [52,34,53], and the adopted representation models strongly depends on the involved technologies [14]. Within these platforms, in particular, users are forced to compose trigger-action rules with a unique, *vendor-centric* abstraction, with which they must specifically refer to every single device or online service needed to execute the intended behaviors. The way a system is presented to the users, i.e., the *abstraction*, may however have a profound impact on users' mental models and expectations. Here, while the *vendor-centric* abstraction allows users to have a fine-grained control, it forces them to define several rules to personalize the IoT ecosystem, i.e., every device or online service needs to be programmed in a specific way. Furthermore, it requires users to know in advance any involved technological detail, e.g., the manufacturer or brand of all the involved "things." Take John, for example:

> *John, a manager of an important company, is always hot, especially in summer. He loves air conditioning, and he would like to set a low temperature wherever it is possible. At home, John has an intelligent Nest thermostat that he controls through his Android smartphone. John goes to work by car. There, all the offices are equipped with a Samsung smart air conditioner.*

Even in such a simplified scenario made only of physical devices, John has to define several rules to reach his comfort goal, at least one for his home, one for his office, and one for his car, even if they perform the same logical operations (i.e., set a specific temperature when he enters a place). Furthermore, he has to be aware of every single technology he may encounter before creating such rules (e.g., Nest, Samsung, etc.), to choose the right one for each rule. In addition, even with an authorization, John will not be able to define similar rules for unknown places or "things" (e.g., his friend's car). As a consequence, little social and practical benefits of End-User Development in the IoT have emerged so far [28].

---

[1] `https://ifttt.com/`, last visited on October 15, 2019

[2] `https://zapier.com/`, last visited on October 15, 2019

Researchers tried to overcome the aforementioned issues from a technological point of view, i.e., by focusing on the TAP platforms: they designed new representation models [17], composition paradigms [7], and debugging tools [15,6]. In this paper, we take a step back to focus on the user, with the aim of understanding what are the gaps between trigger-action programming and the *intentions* of the users. In particular, we build on the idea that other abstractions besides the *vendor-centric* one are possible and desirable, and that the growth of end-user personalization of IoT ecosystems may depend on their identification. By better understanding how people would personalize their devices and services – which abstractions they would adopt and in which contexts – we could better adapt EUD solutions to end-users' needs.

For this purpose, we report on the results of a 1-week-long diary study in the style of a contextual inquiry with 24 participants. The study consisted of two home appointments and a period of time in which participants were free to note down on a diary trigger-action rules arising during their daily activities. In the study, we encouraged participants to think of use cases, both regarding their different physical contexts and their virtual world, in which they would like to personalize the behaviors of their devices and services, and we gathered qualitative data about the current acceptance of IoT personalization. Furthermore, we collected more than 200 trigger-action rules composed by the participants during one week of daily activities. The main contribution of our study are the following:

- We demonstrate that users would define triggers and actions by adopting **different abstractions**. Participants of our study, particularly in the case of programming experts and tech-enthusiasts, used a *device-centric* abstraction to personalize different connected entities, ranging from domestic appliances to car accessories. Through an *information-centric* abstraction, instead, participants went beyond physical devices by shifting their focus to the underlying information, e.g., to personalize their personal plans and appointments, news, and messages. Participants also envisioned their direct involvement in trigger-action rules by defining *people-centric* behaviors. With triggers such as *"when I enter home in the evening"* or *"if me and my friends have free time"*, for example, they explicitly positioned themselves (and other people) *inside* the personalization. On the one hand, this highlights a huge gap with the contemporary *vendor-centric* abstraction, and confirms the need of taking into account the social dimension in the personalization process [4]. On the other hand, such a *people-centric* abstraction points to more intelligent platforms that are able to take into account users' habits and preferences and adapt the actual run-time execution of trigger-action rules, e.g., through solutions based on Artificial Intelligence (AI).
- We show that end-users' needs are not restricted to the smart home context, only, but also open to other smart environments and the "online" world as well, and we demonstrate that the adopted abstraction may depend on the **context** in which the personalization is introduced. Starting from the abstractions used by the participants, we classified the recorded trigger-action

rules in different types, and we analyzed whether different contexts were personalized with different types of rules. We found, for example, that participants mostly used *people-to-information* rules for their health and wellbeing, by connecting a trigger that directly involves the individual to an action for obtaining or manipulating an information. In the smart home context, instead, participants extensively used *device-to-device* rules to customize the joint behavior of different devices or systems. Furthermore, participants used *information-to-information* rules to personalize information when the context was the city or their "online" world.

- We discuss the results by identifying new **design opportunities** to improve the relationship between the Internet of Things, personalization paradigms, and users. Adapting EUD interfaces to different abstractions may reduce the gap between expectations and reality, thus breaking down barriers and increasing the adoption of EUD solutions for an effective personalization of IoT ecosystems. In parallel, we reflect on the role of AI in adaptive platforms for IoT personalization, e.g., in terms of technical challenges and users' acceptance.

2 Background & Related Works

Our study lies at the intersection of End-User Development (EUD) and the Internet of Things (IoT). In this section, we first contextualize our research by describing existing approaches and paradigms for EUD, with a focus on Trigger-Action Programming (TAP). Then, we highlight the issues and challenges that characterize end-user personalization of IoT ecosystems. Finally, we review previous works that explore personalization through the lens of abstraction, by highlighting the contributions that inspired and informed our work.

2.1 End-User Development in the Internet of Things

End-User Development has been defined by Lieberman et al. [38] as *"a set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify or extend a software artifact."* One of the first works in this domain is iCAP [29], a visual, PC-based, and rule-based system for building context-aware applications that does not require users to write any code. To compose a context-aware application with iCAP, users employ the trigger-action approach: they drag components (i.e., devices, locations, time, etc.) either in a "situation" section (*if*) or in an "action" area (*then*). Starting from iCAP, EUD approaches and methodologies have been extensively explored in different contexts. Danado and Paternò, for instance, propose Puzzle [23], a mobile framework which allows end users without IT background to create, modify, and execute applications. In the same mobile context, Namoun et al. [43] uncover the practices of mobile users in respect to developing software apps using mobile devices, and propose a preliminary theoretical model to predict the uptake of software development using mobile devices. Other works, instead, stem from the (now discontinued) Yahoo Pipes to build and explore tools that use formula languages and/or visual programming for data transformation and mashup [37,50,24]. The book of Daniel and Matera [24], in particular, provide a comprehensive overview of core concepts, basic technologies and architectural patterns related to mashup development. Another popular context in which EUD has been extensively studied is the smart home. End-user personalization, in fact, is a key factor in smart home applications [44], and different previous works analyze [52,7] or propose [54,26] tools and approaches to customize smart home environments.

With the technological advances we are confronting today, the EUD vision becomes even more relevant. The Internet of Things (IoT), in particular, had changed the way end users use the Internet, as well as mobile and sensor-based devices, and people are increasingly moving from passive consumers to active producers of information, data, and software [42]. Users can nowadays access new building blocks and tools, analogously to what happened with blogs and wikis during the early phases of the Web [21]. Therefore, EUD paradigms supporting personalization of connected entities, both physical and virtual, are needed [28]: as largely recognized in the literature [31,19,3], EUD methodologies are a viable way for letting users customize their systems to support personal, situational needs. Following this demand, it is not surprising that, in the last years, several cloud-based platforms to support non-technical users in personalizing IoT devices and online services have been proposed. The core idea is to empower users to take advantage of *ecosystems of interoperable smart objects and services* [4], by letting them combine flexibly, i.e., according to their situational needs, the behavior of different entities, e.g., services, data sources, and sensors [28]. TAP platforms such as IFTTT and Zapier have become popular [34] as they offer very easy and intuitive paradigms to synchronize the behavior of devices and applications [39]. Through Web editors, users can define IF-THEN rules, i.e., they can define sets of desired behaviors in response to a specific event. Trigger-action programming is indeed one of the most popular programming paradigm adopted in EUD: it has been largely used for introducing personalization in different contexts, e.g., in the smart home [52,26], for the specification of context-aware applications [29,12] and for web development [25]. Furthermore, it offers a very simple and easy-to-learn solution for creating IoT applications, according to Barricelli and Valtolina [4]. By defining trigger-action rules, in particular, users can connect a pair of devices or online services in such a way that, when an event (the *trigger*) is detected on one of them, an *action* is automatically executed on the second.

For these reasons, in our study, we chose the trigger-action programming paradigm to explore end-users' abstractions in personalizing their IoT ecosystems.

2.2 Personalizing the IoT: Issues and Challenges

Given the popularity of trigger-action programming, we decided to investigate whether there exist gaps between the abstractions offered by TAP platforms and those that end users would adopt. Indeed, existing TAP platforms for personalizing the IoT present their own set of issues. The expressiveness and understandability of IFTTT rules have been criticized since they are rather limited [52,34,53]. Huang and Cackmak [34], for example, report the results of two user studies to understand the impact between different trigger and action types in IFTTT. Their findings reveal inconsistencies in interpreting the behavior of trigger-action programming. Furthermore, cloud-based platforms like IFTTT and Zapier work with well-known IoT devices, only, previously associated to a specific user. Users are indeed forced to compose trigger-action rules with a unique, *vendor-centric* approach, with which they must set all the specific technological details needed to execute the intended behaviors. This clearly poses interoperability challenges: with such a "low-level" abstraction, the user experience with such interfaces is put to a hard test. First, users are forced to know in advance any involved technological detail, and they have to define several rules to program their IoT ecosystems, i.e., every IoT device and online service needs to be managed separately. For instance, a user cannot create a trigger-action rule that can be applied to all her connected lamps, unless they are equally branded, nor to other kinds of devices that may provide interior lighting. Furthermore, contemporary rules cannot include friends' or family's, and users cannot define similar rules for yet unknown places or "things" [17]. In the forthcoming IoT world, however, new "things" will not always be knowable a priori [55] but they may appear and disappear at every moment depending on the user location, e.g., as with public services in a smart city. As a result, little social and practical benefits of End-User Development in the IoT have emerged [28].

A number of previous works try to overcome these issues from a technological point of view, i.e., by developing new features and tools to be used in TAP platforms. Baricelli and Valtolina [4] propose an extension of the trigger-action paradigm that incorporates recommendation systems, other users, and the social dimension. Brich et al. [7] report on the comparison of two different notations, i.e., rule-based and process-oriented, in the smart home context, showing that trigger-action rules are generally sufficient to express simple automation tasks, while processes fit well with more complex tasks. Akiki et al. [1] present ViSiT, an approach that allows end users to specify transformations on IoT objects that are automatically converted into underlying executable workflows. Desolda et al. [28] report on the results of a study to identify possible visual paradigms to compose trigger-action rules in the IoT, and present a model and an architecture to execute them. Ghiani et al. [32] propose a method and a set of tools for end users to personalize the contextual behavior of their IoT applications through trigger-action rules. Corno et al. [15], recently, introduced EUDebug, a system that enables end users to debug and simulate the run-time behavior of their trigger-action rules, to be warned about possible problems that may arise.

Differently from the described previous works, where the focus is on the underlying tools, notations, and/or visual programming paradigms, we try to overcome the issues of contemporary TAP platforms by exploring a different approach, i.e., we focus on the adopted abstractions. By investigating which abstractions end users would adopt and in which contexts, in particular, we aim at going beyond the contemporary *vendor-centric* approach to break down barriers and increase EUD adoption.

2.3 Users and Personalization Abstractions

While IoT ecosystems may take many forms, an EUD solution to personalize them needs to *abstract* the involved devices and services to provide interfaces for end users. These abstractions are typically chosen by designers to guide the users in interacting with and understanding the system. The abstractions we use, however, can create a gap between expectations and reality that prevent adoption [13]. Since people can use multiple and even potentially conflicting models to explain various aspects of a system [45], we claim that multiple abstractions besides the *vendor-centric* one are possible and needed to empower users personalize their IoT ecosystems, and that such abstractions may depend on the context in which the personalization is introduced.

Only few recent works explore the personalization of IoT ecosystems through the lens of abstraction. By investigating the feasibility of letting users customize theit smart homes via trigger-action rules, Ur et al. [52] found that the way users express triggers ranges from events related to sensors to more abstract behaviors that involve multiple devices. In their study, the authors provided the participants with a hypothetical scenario where they had "*a home with devices that are Internet-connected and can therefore be given instructions on how to behave*," thus suggesting a strongly device-oriented abstraction of the system. Since the IoT can be viewed as a complex network of interconnected devices, designers and HCI researchers typically assume a device-oriented abstraction to be presented to the user. Many examples of previous works that stem from such an assumption can be found in the literature [36,46,2]. By exploring triggers and actions that go beyond devices, Corno et al. propose EUPont [17], an ontological representation of End-User Development in the IoT for creating context independent IoT applications based on the users' final goals. Instead of turning on a Philips Hue lamp or opening the bedroom's blinds, for example, with EUPont users can directly ask the system to *illuminate* the room.

Despite the aforementioned works, it still remains unclear which abstraction users would prefer, and whether such an abstraction depends on the context in which the personalization is introduced. The work of Clark et al. [13], however, demonstrates that the way a system is presented to the user can have a priming effect on the initial mental models formed by users, thus influencing how they update their understanding of it based on newly-acquired knowledge [5]. Through a questionnaire submitted to more than 1,500 Mechanical Turk users, in particular, they found that, in the smart home context, users' mental models and the resulting operations that they attempt are heavily affected by the abstractions used to present the system. Starting from this finding, which highlights how critical it is to consciously choose abstractions in the design phase, we decided to explore alternatives to the contemporary *vendor-centric* abstraction *in-the-wild*, by letting the participants of our study freely note down on a diary trigger-action rules arising during their daily activities, at home and outside.

3 Diary Study

To explore end-users' abstractions in personalizing their IoT ecosystems, we devised a diary study in the style of a contextual inquiry related to the composition of trigger-action rules. Although diary studies suffer from the problem that they are tedious for the recorder, they have high ecological value as they are carried out *in situ*, in the users' real environments [22], and they can offer a vast amount of contextual information without the costs of a true field study [8]. We collected data from 24 participants. The study was carried out in July 2017, and was composed of two personal home appointments, separated by a week in which participants freely noted down trigger-action rules arising in their daily activities. All home appointments were audio-recorded. In case of households with more than one inhabitant, the two appointments were conducted separately. All the instructions given to the participants and the collected data were in Italian, their mother tongue. We then translated all the material and results in English for the purpose of this paper. To motivate participants for the study, we drew a prize worth more than 250 € among those who accepted. In this section, and we describe our research questions, the participants, the tools used for the rules composition, and the study procedure.

3.1 Research Questions

We summarize the goals of the study through the following research questions:

- **RQ1 - Needs:** Which physical devices and online services would end users personalize in their daily life?
- **RQ2 - Abstraction:** Which abstractions would end users adopt to compose trigger-action rules?
- **RQ3 - Delegation:** Are users aware of the implications of using a particular abstraction, and which degree of automation would users accept?

3.2 Participants

Table 1 shows the details of the study participants. Due to the sensitive nature of the study, consisting in multiple home visitations, we recruited the participants from our social circle through direct e-mails and messages. We tried to balance the population with respect to the following characteristics: age, gender, living situation, and occupation. The mean age of the participants (15 male and 9 female) was 31.71 years ($SD = 11.47$, $range = 19 - 57$). As reported in Table 1, 18 participants lived in a shared household, i.e., in couple or with more than 2 other inhabitants, while the remaining 6 participants lived alone. The participants' occupations ranged from computer engineer to farmer, thus reflecting a very varied population.

We also asked participants to answer some initial questions about their technological affinity. On a Likert-scale from 1 (Very Low) to 5 (Very High), participants stated their level of technophilia ($M = 3.83$, $SD = 1.09$) and programming experience ($M = 1.58$, $SD = 0.92$).

Table 1. Overview of the 24 participants and their characteristics.

Id	Age	Gender	Occupation	Living Situation	Technophilia	Programming Experience
1	20	F	Student	Shared	4	1
2	26	F	Student	Shared	2	1
3	28	M	Clerk	Alone	4	2
4	24	M	Factory worker	Alone	3	1
5	28	F	Architect	Alone	5	1
6	19	M	Student	Shared	5	3
7	44	M	Construction worker	Shared	2	1
8	43	F	Teacher	Shared	2	1
9	52	F	Teacher	Shared	4	1
10	21	F	Student	Shared	4	1
11	32	M	Environmental engineer	Alone	3	2
12	25	M	Student	Alone	5	2
13	22	M	Farmer	Shared	5	1
14	26	M	Farmer	Shared	4	1
15	25	F	Office worker	Shared	2	1
16	22	M	Airplane pilot	Alone	4	1
17	32	F	Teacher	Shared	5	1
18	28	M	Factory worker	Shared	4	2
19	42	M	Entrepreneur	Shared	5	2
20	35	F	Office worker	Shared	3	1
21	55	M	Office worker	Shared	5	2
22	57	M	Agricultural consultant	Shared	3	2
23	36	M	Computer Engineer	Shared	4	5
24	19	M	Student	Shared	5	2

3.3 Rule Notation and Composition Kit

To allow participants to define IoT personalizations in their daily lives, we created a *composition kit* with which users could freely note down trigger-action rules arising during their daily activities at home or outside. We chose to adopt the trigger-action paradigm due to its simplicity and its popularity in the context of End-User Development [4]. As done by Brich et al. [7] in the smart home context, we chose to build a pen and paper kit to encourage creativity without artificially restricting the elicitation process to a specific user interface. As reported by Rodden et al. [48], there are many advantages in using sketches since they are: a) *disposable*, i.e., they encourage participants to criticize aspects without fear of commenting a researcher's work, as could be the case with fully developed interfaces; b) *minimalist*, i.e., they help focus on the core aspects of the research questions, without introducing technological constrains; c) *explorative*, i.e., they allow participants to come up with creative solutions; and d) *ambiguous*, i.e., they allow participants to appropriate the sketched content more easily and make it their own. The *composition kit* consisted on a home-made book and a pen (Figure 1).

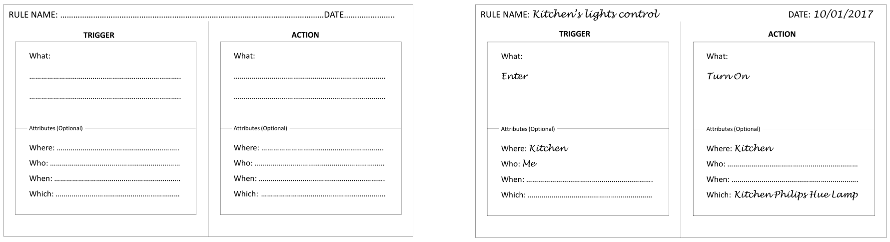
Fig. 1. The kits we used for the diary study, composed of a booklet and a pen.

To express IoT trigger-action rules in the study, we defined a *rule notation*. To focus on the research questions without introducing unnecessary complexity for end users, we adopted the simplest form of the trigger-action programming approach, i.e., each rule contains exactly one trigger and one action. However, we allowed participants to enrich each trigger and actions with multiple restrictions, as suggested by Ur et al. [52]. In particular, similarly to Desolda et al. [28], we followed a *5W* model for defining triggers and actions. The original *5W* model is adopted in several domains, such as journalism and customer analysis, to analyze the complex story about a fact through the following keywords: **W**hat, **W**ho, **W**hen, **W**here, and **W**hy. We adapted the model in our notation by specializing the meaning of each keyword to our domain, and by replacing the **W**hy keyword with **W**hich keyword (Figure 2(a)). In particular, **W**hat is used for describing the trigger or the action, while **W**ho, **W**hen, **W**here, and **W**hich are used as social, temporal, spatial, and technological constraints, respectively.

To compose trigger-action rules, the booklet contained 20 pages with the *rule notation* template (Figure 2(a)), a brief manual, and a rule example (Figure 2(b)). The following statements, which describe the *rule notation*, were reported on the manual:

- The adopted notation is rule-based: each behavior is defined in the *if THIS then THAT* form.

- The if-clause (named trigger) represents an event to be observed, while the then-clause (named action) defines an action to be executed after the verification of the if-clause.
- Each rule contains one single trigger and one single action.
- Each trigger and action must be defined by filling the following template:
  - What (mandatory): the trigger to be observed, or the action to be executed. The *what* can be defined using the preferred level of detail, without any restriction.
  - Where (optional): the physical location on which the trigger will be checked, or the action must be executed, e.g., the kitchen, the office.
  - Who (optional): the source/target person or people of the trigger or the action, e.g., the recipient of a message.
  - When (optional): a temporal restriction for the trigger or the action.
  - Which (optional): an indication of the devices or services to be used for reproducing the trigger or the action.
- Each rule must be defined in a different sheet of paper.
- Each sheet of paper already contains the following elements to be filled:
  - A field in the upper left corner for a meaningful rule-name.
  - A field in the upper right corner for the date.
  - A trigger template on the left column.
  - An action template on the right column.



RULE NAME: .... DATE.....................
TRIGGER
What: ....
Attributes (Optional)
Where: ....
Who: ....
When: ....
Which: ....
ACTION
What: ....
Attributes (Optional)
Where: ....
Who: ....
When: ....
Which: ....

RULE NAME: Kitchen's lights control
DATE: 10/01/2017
TRIGGER
What: Enter
Attributes (Optional)
Where: Kitchen
Who: Me
When: ....
Which: ....
ACTION
What: Turn On
Attributes (Optional)
Where: Kitchen
Who: ....
When: ....
Which: Kitchen Philips Hue Lamp

(a) (b)

Fig. 2. The content of the booklet. Figure (a) shows the template reported on the 20 book's pages for composing trigger-action rules. The fields to be filled follow our *rule notation*. Figure (b) show the rule example reported in the the booklet. We intentionally used different abstractions to avoid bias: the trigger is very abstract, because it does not include any devices or services, while the action specifically refers to a Philips Hue lamp.

3.4 Study Procedure

The study procedure we devised was composed of 3 phases (Figure 3). In a first appointment (Initial Appointment), we visited participants in their home to introduce the study and give them the *composition kit*. Then, we instructed participants to bring the kit with them for an entire week, with the aim of composing IoT trigger-action rules on the basis of their daily activities and experienced contexts (Composition Phase). Finally, in a subsequent appointment (Final Appointment), we collected the recorded rules, and we performed a semi-structured interview.

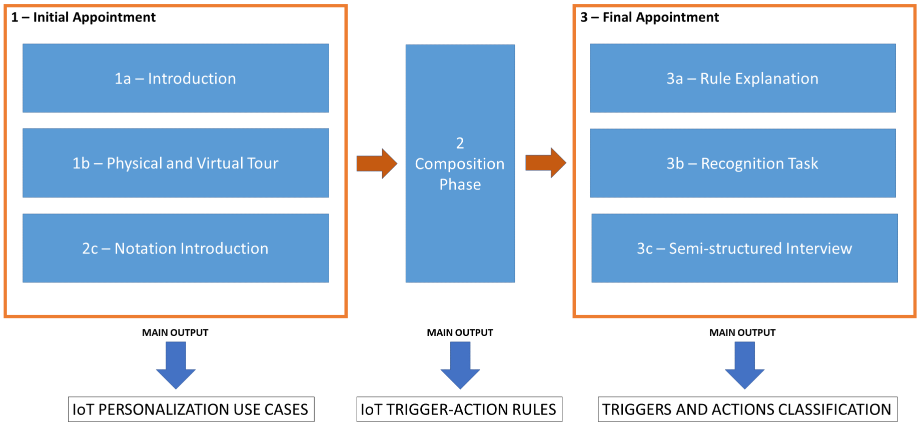


Fig. 3. The procedure we adopted for the study, composed of two personal appointment and a composition phase of one week. The figure reports the main output we retrieved from each phase. In the Initial Appointment we collected use cases of IoT personalization. In the Composition Phase we collected IoT trigger-action rules. The Final Appointment allowed us to classify triggers and actions in terms of level of abstraction.

Initial Appointment To start the study, we set up a first appointment at the participants' home. The appointment took about 30 minutes. We divided it in the following three distinct phases:

1a) **Introduction.** Participants were introduced to the general idea of EUD in the IoT context. First, the IoT paradigm was introduced, along with some examples of IoT devices and services. Then, participants were taught about the trigger-action programming approach. We concluded this phase with three examples of IoT trigger-action rules in various environments, paying attention to use different abstractions.

1b) **Physical and Virtual Imagination Tour.** To spark the participants creativity toward EUD in the IoT and initially explore which devices and services users would personalize (**RQ1**), we invited them to formulate use cases of IoT personalization in an unrestricted manner, disregarding any technological and notation constraints. In such an "imagination" tour, we asked participants to think about their daily activities, both in their physical world (e.g., the home, the workplace) and virtual world (e.g., social networks, messaging services). Participants could state whatever behavior they would want to personalize, even if not feasible. In this phase, the experimenter could ask participants if they would find it useful to personalize a use case, but could not tell them what personalization could be useful.

1c) Notation Introduction. Participants were then introduced to the *rule notation*. To familiarize with the notation, participants were requested to represent in the trigger-action form two of the use cases defined in phase 1b (one from the "physical tour", and one from the "virtual tour"). Finally, the *composition kit* was given to the participants: they were instructed to take with them the kit in all their daily activities, and to record as many trigger-action rules as possible until the second appointment one week later.

Composition Phase In the week between the two appointments, participants were encouraged to use the *composition kit* to compose trigger-action rules during their daily activities. As reported in the Results section, participants composed a total of 233 trigger-action rules referring to both physical and virtual ecosystems.

Final Appointment To end the study, we revisited participants in their home in a second appointment. It took about 30 minutes, and was composed of the following phases:

3a) Rules Explanation. At the beginning of the second appointment, participants were asked to show the trigger-action rules they had defined by explaining their mental process retrospectively. The focus of this phase was to understand whether the noted down trigger-action rules were correctly representing the ones participants envisioned, and whether the abstraction the users had in mind was consistent with the one that emerged from the noted down rules (**RQ2**).

3b) Recognition Task. To further investigate whether the participants were aware of the abstractions they used, we exposed them to a recognition task. Participants, in particular, were firstly introduced to the *vendor-centric* abstraction through some examples of rule composition on the IFTTT platform. Then, they were asked to analyze their collected rules and state if the corresponding triggers and actions followed the *vendor-centric* abstraction or not.

3c) Semi-structured Interview. The last part of the study consisted in a semi-structured interview. The following questions were used to investigate **RQ3**:

- **Q1.** In your opinion, is there a reason why you used different abstractions in your triggers and actions?
- **Q2.** In case of generic triggers or actions, e.g., those behaviors that cannot be modeled by the contemporary *vendor-centric* abstraction, would you accept an intelligent system that automatically decides the actual behavior to be reproduced, e.g., by choosing final devices and services?

- **Q3.** By knowing the differences between the abstractions that you used in your rules and the *vendor-centric* abstraction, would you now change something in the triggers and actions you defined?

4 Results

Thanks to the study, we collected in total 109 freely formulated use cases of IoT personalization, 233 trigger-action rules, and 25 hours of audio recording. Qualitative analysis of the audio recordings was conducted by two coders in an iterative coding process. Inter-rater reliability was determined using Fleiss' Kappa coefficient. We settled on four major categories, i.e., *Potential and Acceptance*, *Abstractions*, *Contexts*, and *Delegation*, that we use to organize the result presentation. We support qualitative outcomes with quantitative results. To further explore our data, we also divide participants in four groups, on the basis of a) their programming expertise, i.e., by considering experts those participants that declared a programming experience greater than 3 and b) their enthusiasm towards technology, i.e., by considering enthusiasts those participants that declared a technophilia greater than 3 (in both cases, out of a Likert-scale of 5). Results are shortly discussed where necessary, while insights and new design opportunities are presented in the next section.

On average, each participant contributed to the study with 9.71 rules ($SD = 3.74$). Programming experts tended to record more rules with respect to participants with limited programming experience ($M = 11.34$, $SD = 4.04$, *vs.* $M = 9.47$, $SD = 3.75$, respectively). The technophilia, instead, did not affect the number of collected rules: both tech-enthusiasts and non enthusiasts recorded, on average, a very similar number of rules ($M = 9.81$, $SD = 3.43$, *vs.* $M = 9.50$, $SD = 4.57$, respectively).

4.1 EUD in the IoT: End-Users' *Needs*

To investigate which devices and services users would like to personalize (**RQ1**), we considered the use cases emerging in the Initial Appointment, as well as the trigger-action rules collected in the Composition Phase, with the aim of determining clusters. For this purpose, we analyzed the collected dataset by looking for frequency of key words as well as evaluating them qualitatively through open coding. To make sure that related words were not treated separately, the word frequency was based on word stems by using the Porter stemmer [47].

Prominent use cases collected during the Initial Appointment were about lights control (11), web and social network automation (11), and doors and windows control (6). Figure 4, instead, shows a distribution of the devices and services mentioned more than 6 times in the use cases and in the **W**hat and **W**hich fields of the trigger-action rules collected during the Composition Phase.

Not surprisingly [7], the chart shows that lights, doors, windows, and thermostats are the most common connected entities that participants would like to

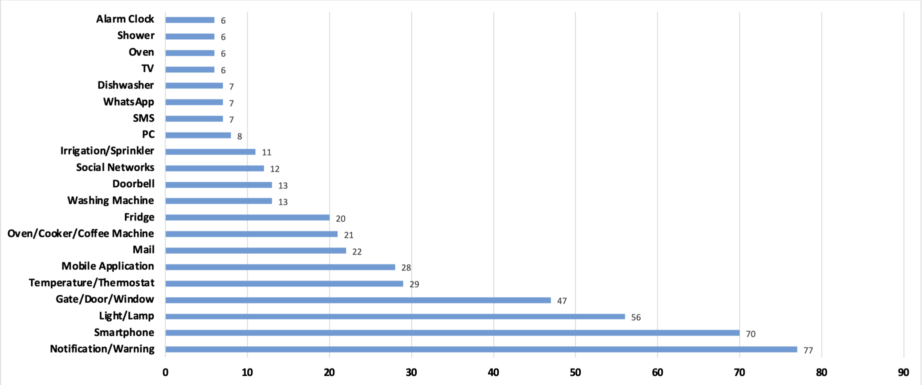


Fig. 4. Distribution of devices and services that participants would like to personalize.

program. However, the graph highlights that end-users' needs are not restricted to the physical devices installed in their smart home. As better detailed later for the **W**here restriction, about 50% of the rules did not include a location, and were intended to work in *any* context. Furthermore, beside the home, a considerable number of rules referred to other smart environments, e.g., the car and the workplace, and to the "virtual" world, e.g., mail and social networks. Moreover, the smartphone was mentioned 52 times out of 70 use cases and trigger-action rules which do not belong to the smart home context. Examples of such use cases and rules include *"I would like something to block the smartphone when I am driving"*, *"if there are important news from the world, then show them on my smarphone"*, or *"the smartphone should notify me when I exceed a certain weight"*. Several (77) use cases and trigger-action rules, in particular, aimed at personalizing notifications/warnings. In the majority of these cases (57), participants referred to notifications/warnings without specifying any physical medium, e.g., by using generic triggers like *"when I receive Facebook notifications"* or generic actions like *"alert me."* In the remaining cases, instead, participants referred to their smartphone notification system (15) or included both the smartphone and the PC (5). A participant, for example, came out with the following use case: *"I would like a smart notification filter on my smartphone, something that automatically blocks undesired spam and advertisements."* Another participant, instead, personalized smartphone notifications to be warned in case of problems: *"if the dog runs away from home, then call me and send me a notification on my smartphone."* Also notifications and warnings were mentioned many times for contexts different than the smart home (56 out of 77). An example is *"if my friends are near me, then send me a notification."* Such a variety of contexts confirms the need of exploring the End-User Development in a wider scenario.

By further analyzing the graph, we can see that participants referred to devices and services with different levels of detail: in some cases, e.g., the smartphone, participants did not include any manufacturers or brands, while in other cases, e.g., WhatsApp, they provided the specific product name. Between all the use cases and trigger-action rules, only in 18 cases participants mentioned specific sensors, e.g., motion sensors, humidity sensors, etc. This confirms that end users tend not to mention sensors directly [29,51,52], and preliminary demonstrates that the contemporary *vendor-centric* abstraction is not sufficient to model all the behaviors envisioned by the users. For what concerns the other optional restrictions that users could specify in the trigger-action rules, we found the following results:

Where: 127 triggers (54.51%) and 74 actions (31.76%) included a location restriction. The most prompted location was the home (132 times), followed by the car (66 times), the home garden (33 times), and the workplace (15 times).

Who: 102 triggers (43.78%) and 64 actions (27.47%) included a people restriction. Participants explicitly defined 106 behaviors, i.e., triggers or actions, for themselves, while 19 behaviors were defined for their family, and 10 behaviors for their pets.

When: 55 triggers (23.61%) and 35 actions (15.02%) included a temporal restriction. The temporal restriction was used in many different ways. In 15 cases, the restriction specified a precise hour of the day. In 30 other cases, the restriction specified a generic instant/period, e.g., *in the evening*, *in the summer*. In the remaining 45 cases, the restriction was used to specify a contextual constraint on the trigger or on the action, e.g., *I'm driving*, *I'm working*.

The usage of the restrictions further shows that end users are not only interested in the smart home context, but in other areas such as the car and the workplace. Furthermore, users would like to define rules that can be adapted for different subjects, e.g., for all the family, and in many cases they suppose that an EUD system should know their habits and preferences, e.g., in the case of the temporal/activity restrictions as *in the evening* or *when I'm working*.

To understand how end users perceive the Internet of Things, and to investigate their acceptance to IoT personalization, we qualitatively analyzed the audio recordings of the two appointments. The majority of the participants (20) were excited about the topic of the study. On the contrary, 4 participants explicitly declared their opposition to IoT personalization, and in general to technology. A participants said *"I prefer to do things manually,"* while another said *"I would like to live without so much technology."* As a possible confirmation of such a trend, the rule *"if I use too much the social networks, then block them"* was very common, and was defined by 5 other participants.

Another aspect we extracted from the analysis is that sometimes End-User Development is not an easy task. Two participants declared they struggled to reason in the "trigger-action" way. A participant said *"sometimes I were not able to define what was the event, and what was the action."* Other 2 participants pointed out that the current technology is so advanced that it is difficult to think of innovative scenarios. Furthermore, another participant said *"it is hard to be satisfied with such rules because you can easily forget what you defined."*

4.2 Personalizing the IoT Through Different *Abstractions*

Triggers and Actions Abstractions. To explore which abstractions end-users would adopt in personalizing their IoT ecosystems (**RQ2**), we firstly analyzed the recorded triggers and actions, along with the participants' explanations collected in the Final Appointment, with the aim of determining clusters. To classify triggers and actions, in particular, we adopted the categorization proposed by Clark et al. [13] in the smart home context, according to which a personalization may fall in one these categories:

Device-centric A personalization whose subject is the physical medium with which it is executed. In our context, device-centric triggers and actions specified a device either directly in the **W**hat field or in the **W**hich field. A device-centric trigger, in particular, represents an event that is recognized by a physical object, while a device-centric action is the execution of an automatic behavior on a physical object. Participants, for example, used a device-centric abstraction to detect when the *garage door* closes (P1), to monitor the *car's* speed (P15), or to discover when there is an electrical failure in the *home lighting system* (P6).

Information-centric A personalization whose subject is the underlying information, regardless of the physical medium with which it is manipulated. In our context, information-centric triggers and actions specified such an information either directly in the **W**hat field or in the **W**hich field. A data-oriented trigger, in particular, represents an information that becomes available, while a data-oriented action is an information to be automatically obtained. Participants, for example, used an information-centric abstraction to monitor *their university exams* (P2), to detect when a *dangerous web site* has been visited (P14), or to manage *Facebook's notifications* (P12).

We independently labeled the trigger and the action parts of each collected rule. When rules were expressed in an ambiguous way, we used the qualitative data collected during the home appointments. Then, we discussed disagreements by reaching a consensus (Cohen's kappa = 0.93, SD = 0.07). During such a process, we found a large group of triggers that did not follow a device-centric nor infomation-centric abstraction. While these triggers resembled the "fuzzy triggers" discovered by Ur et al. [52] in the smart-home context, they also shared an additional characteristic, i.e., all of them envisioned a direct involvement of the participant in the personalization. We therefore defined an additional abstraction:

People-centric A personalization where users, their actions, and/or feelings are at the center of the interaction, independently of any physical and virtual medium. In our context, people-centric triggers had typically an empty **W**hich field, and they explicitly mentioned an individual or a group of individuals either directly on the **W**hat field or in the **W**ho field. Participants, for example, used a people-centric abstraction to trigger an event whenever *they arrive* at home (P8), to monitor *family members* (P20), or for more futuristic ideas, e.g., to detect when *they are hungry* (P20).

Table 2 reports some examples of triggers and actions for each used abstraction.

As shown in Figure 5, participants demonstrated to prefer the *device-centric* abstraction when defining triggers (101 times), but they consistently used the *information-centric* and the *people-centric* abstraction, too (70 and 62 times, respectively). In defining actions, instead, participants adopted the *device-centric* and the *information-centric* abstractions in a similar way (120 and 113 times, respectively).

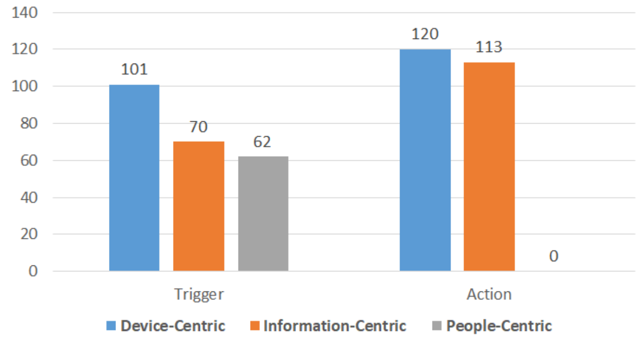


Fig. 5. The distribution of the abstractions used by the participants in triggers and actions.

To further investigate which abstractions participants used, we explored differences between participants' groups, i.e., programming experts *vs.* non experts and tech-enthusiasts *vs.* non enthusiasts. We did not find any significant statistical difference between participants, but interesting qualitative trends emerged. Results of such an investigation are reported in Table 3 and discussed hereafter.

On average, each participant defined 4.21 *device-centric* triggers ($SD = 2.90$), 2.83 *information-centric* triggers ($SD = 2.16$), and 2.58 *people-centric* triggers ($SD = 2.02$). Looking at triggers, a similar trend emerge when considering both programming experience and technophilia. By defining on average 6.00 ($SD = 5.29$) and 4.63 ($SD = 2.70$) *device-centric* triggers, respectively, programming experts and tech-enthusiasts demonstrated their preference towards including devices in their personalizations. Furthermore, they specified devices more often than non-experts and non-enthusiasts. On the contrary, people with limited programming experience and enthusiasm used the 3 different abstractions in a similar way. Non-enthusiast participants, for example, recorded on average 3.36 *device-centric* triggers ($SD = 3.29$), 2.87 *information-centric* triggers ($SD = 2.03$), and 3.12 *people-centric* triggers ($SD = 2.23$). Such results seems to suggest that people that already know how to program and love technology prefer to maintain control over their IoT ecosystems. In the final appointment,

Table 2. Some examples of triggers and actions for each used abstraction. The examples provide an intuitive feel for how the triggers and actions differ between abstractions: *device-centric* triggers and actions explicitly refer to a device or a system, e.g., the lights, while in *information-centric* behaviors the focus is on the information, e.g., the user position. Finally, *people-centric* triggers imply a direct involvement of the user in the rule, with events that are strictly related to the individual, e.g., arriving in a place. The reported triggers and actions have been rephrased for the sake of readability.

	Trigger	**Action**
Device Centric	"The car break the speed limit" (P15) "There is an electrical failure in the home lighting system" (P6) "The humidity sensor detects that the grass is dry" (P23) "I close the garage door" (P1)	"In the evening, turn the lights of the courtyard on" (P7) "Turn on the irrigation system" (P14) "Limit the car speed for me and my boyfriend" (P15) "Automatically set the table" (P4)
Information Centric	"The university exams are approaching" (P2) "My son is browsing the Web" (P14) "When I receive a Facebook notification" (P1) "When I publish a post on a social network" (P23)	"Send me a notification" (P12) "Recommend me a training program according to my previous activities" (P5) "Order groceries online" (P10) "Send my position to a favorite phone number" (P10)
People Centric	"The workers enter the factory" (P4) "I arrive at home"" (P8) "A family member is home alone" (P20) "When I'm hungry" (P3)	-

Table 3. The table reports how many times on average each participant defined a *device-centric*, *information-centric*, or *people-centric* trigger or action.

		Programming Experience		**Technophilia**	
	All	**Expert**	**Non Expert**	**Enthusiast**	**Non Enthusiast**
Device-Centric Triggers	4.21 (2.90)	6.00 (5.29)	3.95 (2.52)	4.63 (2.70)	3.36 (3.29)
Info-Centric Triggers	2.83 (2.16)	2.67 (2.52)	2.85 (2.17)	2.81 (2.29)	2.87 (2.03)
People-Centric Triggers	2.58 (2.02)	2.33 (2.08)	2.62 (2.06)	2.31 (1.92)	3.12 (2.23)
Device-Centric Actions	5.00 (3.36)	5.00 (1.73)	5.00 (3.56)	4.81 (3.08)	5.37 (4.07)
Info-Centric Actions	4.71 (2.27)	6.33 (2.52)	4.48 (2.20)	5.00 (2.50)	4.12 (1.73)

when asked whether there was a reason for using different abstractions in his triggers and actions, P23 (a computer engineer) said:

> *"I love being specific. Since I know what to do, I would program each device in a specific way."*

When considering actions, instead, differences are less prominent and no explicit trend emerges: independently of their programming expertise and technophilia, participants defined actions both with the *device-centric* and the *information-centric* abstraction.

From the Adopted Abstractions to Different Types of Rules. The abstractions used by participants for defining triggers and actions lead to different types of rules. Table 4 describes the retrieved rule types and presents some examples. From *device-to-device* rules, i.e., rules with both the trigger and the action expressed with a *device-centric* abstraction, to *people-to-device* rules, i.e., rules with a *people-centric* trigger and an *information-centric* action, participants personalized their IoT ecosystems in very different ways.

As reported in Figure 6, the majority of rules were of type *device-to-device* (65). With such rules, participants defined an action over a physical entity to be executed when something happened or was detected by another physical entity. Following the *device-centric* abstraction for both trigger and action, in particular, participants referred to physical "things" ranging from sensors to more complex systems. In the rule:

> *"[If] the tensiometer detects that the soil is dry, the irrigation system is turned on,"*

for instance, P11 mentioned a sensor in the trigger (the tensiometer), while he generically referred to a more complex system in the action (the irrigation system).

Another popular type of rules was *information-to-information* (47). Rules of such a type were recorded with the aim of obtaining or manipulating an information when another information was available, e.g., to get a study plan when

Table 4. The abstractions used by participants for defining triggers and actions lead to different types of rules, from *device-to-device*, where rules involve devices, only, to *people-to-information*, where triggers that directly involve users are used to obtain or manipulate information. The reported rules have been rephrased for the sake of readability.

	Description	**Examples**
Device To Device	Rules to execute an action over a physical entity when something happens or is detected by another physical entity.	"[When] the tensiometer detects that the soil is dry, the irrigation system is turned on" (P11) "[If] I'm using the smartphone while I'm driving, [then] block it" (P3)
Device To Information	Rules to obtain or manipulate an information when something happens or is detected by a physical entity.	"[When] the sensor detects that it's raining, [the system] warns me through a WhatsApp message or a SMS" (P18) "[If] I park the car, save my position on the smartphone" (P12)
Information To Device	Rules to execute an action over a physical entity when a new information is available.	"[When] the weather conditions change while I'm on my car, the car radio starts playing songs that better fit with the new conditions" (P6) "[If] I use social networks for more than x hours, [then] block them on all my devices" (P11)
Information To Information	Rules to obtain or manipulate an information when another information is available.	"[When] my bank account exceeds a threshold, propose me safe financial investments" (P11) "[When] the university exams are approaching, provide me a plan for studying" (P2)
People Io Device	Rules to execute an action over a physical entity when an individual perform a generic action or her conditions change.	"When my daughter is coming home in the weekend, I would her room to be automatically warmed" (P21) "[If] I'm hungry at night, the lock the fridge and the food storage" (P3)
People To Information	Rules to obtain or manipulate an information when an individual perform a generic action or her conditions change.	"[If] my friends and I have free time, propose us something to do through a smartphone notification" (P16) "[When] I enter home in the evening, announce me the tasks I have to do" (P20)

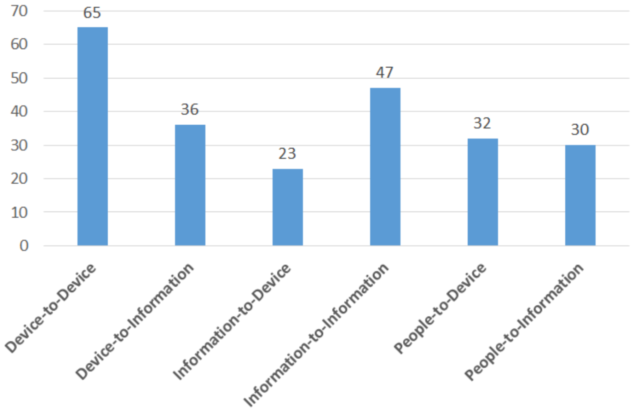


Fig. 6. The distribution of the different types of rules coming from the abstractions used by the participants.

the university exams are approaching (P11). Information was also combined with devices. Through *device-to-information* rules (36), in particular, participants expressed their need of being informed when something happened on a physical entity, e.g., to be notified when a sensor detects that it is raining (P18). Through *information-to-device* rules (23), instead, participants defined an action over a device or a system to be executed whenever a new information was available, e.g., to block devices when they spend too much time on social networks (P11).

In the collected rules, the *people-centric* abstraction was frequently used both for controlling physical entities (*people-to-device* rules, 32) and for obtaining or manipulating information (*people-to-information* rules, 30). A participant, for example, asked to automatically warm a room whenever her daughter is coming home in the weekend (P21), while another would like to receive suggestions about things to do whenever she and her friends have free time (P16).

The qualitative trends emerging from the analysis of the abstractions used in triggers and actions (Table 3) are confirmed when performing the same analysis for each rule type. Table 5, in particular, highlights a connection between *device-to-device*, *device-to-information*, and *information-to-information* rules with programming experience and tech-enthusiasm. Instead, excluding *device-to-device* rules, i.e., the most common type, people with limited programming experience and enthusiasm used all the rule types in a similar way, on average.

4.3 The Right Abstraction in the Right *Context*

Personalizing Different Contexts. As already reported in the previous sections, we found that participants introduced personalizations in different contexts. Figure 7 further details the different contexts in which participants introduced their trigger-action rules. By means of 94 different trigger-action rules,

Table 5. The table reports how many times, on average, each participant defined rules of a given type.

		Programming Experience		**Technophilia**	
	All	**Expert**	**Non Expert**	**Enthusiast**	**Non Enthusiast**
Device-to-Device	2.71 (2.18)	3.00 (2.65)	2.67 (2.18)	2.87 (1.86)	2.37 (2.83)
Device-to-Info	1.50 (1.53)	3.00 (2.65)	1.29 (1.27)	1.75 (1.77)	1.00 (0.76)
Info-to-Device	0.96 (1.04)	0.67 (1.15)	1.00 (1.05)	0.81 (1.11)	1.25 (0.89)
Info-to-Info	1.96 (1.60)	2.33 (2.08)	1.90 (1.58)	2.06 (1.61)	1.75 (1.67)
People-to-Device	1.33 (1.66)	1.33 (1.53)	1.33 (1.71)	1.13 (1.63)	1.75 (1.75)
People-to-Info	1.25 (1.51)	1.00 (1.00)	1.28 (1.59)	1.19 (1.60)	1.37 (1.41)

participants often personalized the behaviors of their home, thus confirming the user's interest in home automation [7].

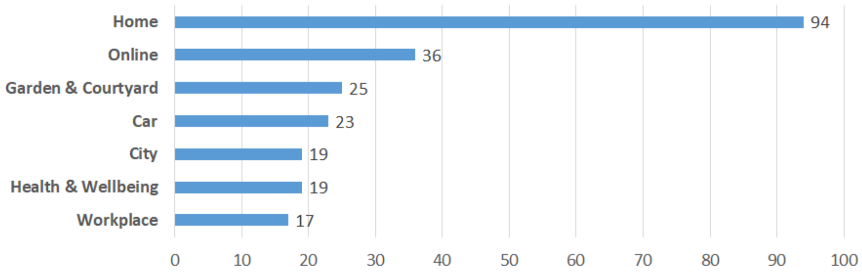


Fig. 7. The contexts in which participants introduced their trigger-action rules. Despite the home remains the most popular context, results show that end-users' needs also involve other environments such as the workplace and the car, and the online world as well.

In 20 cases, in particular, participants defined rules to control home appliances, ranging from the coffee machine to the fridge. In some cases, participants referred to multiple appliances in the same rule. P19, for example, defined the following rule:

> *"if the dishwasher, the washing machine, and the oven are all turned on at the same time, a limiter automatically deactivates other not essential appliances, to avoid failures in the home lighting system."*

This highlights a gap between end-users' mental models and the contemporary *vendor-centric* abstraction. The latter, in fact, typically allow users to program one appliance at a time.

Despite the popularity of the home context, participants also personalized other smart environments, such as their gardens and courtyards (25), their car (23), and their workplace (17). The car, in particular, was mentioned in rules with different purposes. P15 and P19, for example, considered the car as the focus of the personalization, and defined trigger-action rules to personalize the car behavior:

> *"if my boyfriend or I exceed the speed limit by car, [then] decrease the car speed within the limit"* (P15);
>
> *"when I'm driving, adjusts my car's temperature according to the external weather conditions"* (P19).

P3 and P12, instead, used the car as context, while defining rules with a different focus:

> *"[if] I'm driving my car and I'm using the smartphone, automatically block it!"* (P3);
>
> *"[When] I parked the car, save its position on my smartphone"* (P12).

In addition to environments under their strict control, e.g., the home and the car, participants also envisioned rules in the city context (19 times), thus defining triggers and actions that involve environments, devices, and services that could be potentially accessed by all the citizens. The following 2 rules, for instance, were defined to be notified when some events happen around the city:

> *"monitor air pollution and send me a notification on my smartphone when pollution values exceed a threshold"* (P22);
>
> *"[if] there is a nearby car accident, send me a notification"* (P19).

Besides "physical" environments, we found that participants' rules frequently involved their online world (36 times) and their health and wellbeing (19 times). With rules such as

> *"[if] I publish a post on a social, [then] post it on all the other social networks"* (P4);
>
> *"[if] there are interesting local and global news, [then] send me a notification"* (P23);
>
> *"when my car insurance is about to expire, perform a market research on the web"* (P14).

participants personalized social networks, news, and their "online" information in general.

For their health and wellbeing, instead, participants often defined automatic notifications to be received whenever their health parameters changed:

> *"in case of hearth problems, send me a notification"* (P12);
>
> *"when I exceed a certain weight, send me a smartphone notification"* (P15).

Using Different Abstractions in Different Contexts. After identifying the contexts for which participants recorded trigger-action rules, we further studied the relationship between the context and the type of the rule. Our aim was to investigate whether the adopted abstraction depended on the context in which the personalization was introduced. Figure 8 shows how many times (in percentage) participants used a given rule type in a specific context.

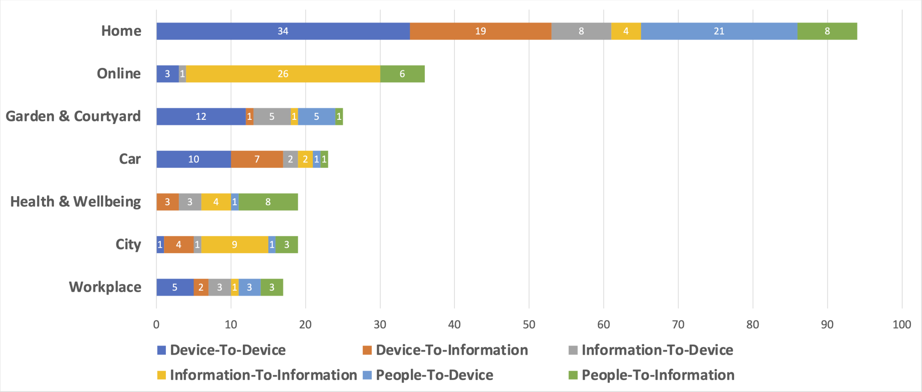


Fig. 8. How many times participants used a given rule type in a context. Looking at the figure, interesting patterns emerge. While *device-to-device* rules were prominent in the home context, for example, in the city context participants were more interested in information rather than devices. Furthermore, participants defined *people-centric* triggers in all the contexts, especially in their workplace, their home, and for their health and wellbeing.

By analyzing the figure, interesting patterns emerge:

- In the home, garden & courtyard, car, and workplace, i.e., the "physical" contexts under their strict control, participants extensively used *device-to-device* rules: 34 cases out of 94 (36.17%) in the home, 12 out of 25 (48%) in garden & courtyard, and 5 out of 17 (29.41%) in the workplace. In such contexts, also *information-centric* and *people-centric* triggers and actions were typically associated with physical devices or systems, while *information-to-information* rules, i.e., personalizations involving information, only, were rarely used: 4 cases out of 94 (4.26%) in the home, 1 out of 25 (4%) in garden & courtyard, and 1 out of 17 (5.88%) in the workplace.
- In the city, participants were more interested in information rather than devices. Only 1 rule our of 19 (5.26%) was of type *device-to-device*. On the contrary, participants defined *information-to-information* and *device-to-information* rules in 47.37% (9 out of 19) and 21.05% (4 out of 19) of cases, respectively. Furthermore, while in the other physical environments the *people-centric* abstraction was frequently associated to devices, in the

city participants preferred *people-to-information* rules (3 rules out of 19, 15.79%).
- Not surprisingly, the *information-centric* abstraction was prominent in the online context, and it appeared in the 91.67% of the related rules in total. *Information-to-information* rules, in particular, were the most common (26 rules out of 36, 72.22%).
- Participants used the *people-centric* abstraction in all the contexts. With the exception of the car (8.7%) and the online context (16.67%), all the other contexts were personalized through *people-centric* behaviors in more than 20% of cases. In the majority of the rules for the health & wellbeing context, for example, participants used *people-to-information* (8 out of 19, 42.11%) and *people-to-device* (1 out of 19, 5.26%) rules. *People-to-information* rules were also frequently used in the workplace (3 rules out of 17, 17.65%), in the online context (6 rules out of 36, 16.67%), and in the city (3 rules out of 19, 15.79%). In other contexts such as the home and the garden & courtyard, the *people-centric* abstraction was more often involved in *people-to-device* rules (22.34% and 20%, respectively).

4.4 Can We *Delegate* an Intelligent System to Execute our Personalizations?

By analyzing the audio recordings of the final appointment, i.e., the participants' explanations of their rules and the debriefing session, we investigated whether participants were aware of the implications of using a particular abstraction (**RQ3**). The used abstractions, in fact, can have an impact on the surrounding environments, especially at run-time. While a trigger defined with the contemporary *vendor-centric* abstraction is monitored via a specific device or online service, for example, this is not true for *people-centric* triggers, as they could be executed and adapted in different ways at run-time. The *"when I enter home"* trigger, for instance, could be monitored through a door that has been opened, or through a security camera detecting movements, among the others.

Abstractions Feasibility. The usage of a diary-based methodology poses questions about the technical feasibility of the composed rules, especially for "generic" triggers and actions, e.g., those expressed with a *people-centric* abstraction. Similarly to previous works exploiting a similar methodology (e.g., [7]), we only observed few futuristic rules (5, 2.15%) impossible to be executed, at least in the near future, with contemporary technology. All of them included very generic *people-centric* triggers, such as *"when I'm hungry"* or *"when I'm curious about something."*

In all the other cases, the recorded trigger-action rules were either executable as they were with the contemporary *vendor-centric* abstraction (143, 61.37%) or adaptable in some ways at run time (85, 36.48%). In particular, we considered as executable all the rules that included *device-centric* or *information-centric* triggers and actions referring to specific devices, systems, or online services.

Instead, we considered as adaptable all the rules with “non impossible” *people-centric* triggers, and all the rules which *device-centric* and *information-centric* triggers and actions that did not refer to specific physical or virtual entities.

To investigate whether participants were aware of the differences between the abstractions they used and the *vendor-centric* one, we analyzed the classification made in the Recognition Task. We found that participants misinterpreted 19 triggers out of 233 (8.15%), e.g., by stating that a trigger followed the contemporary *vendor-centric* abstraction when in fact it was too generic to be linked to a specific device or service. Errors on the action classification were instead 42 out of 233 (18.02%). The number of misclassifications is not negligible and it tells us that in some cases participants used different abstractions unknowingly.

Generic Triggers and Actions at Run-Time. When asked whether they would accept an intelligent system to automatically execute more “generic” triggers and actions, e.g., *people-centric* triggers, 17 participants out of 24 (70.84%) answered yes, at least in some cases. This seems to be partially in conflict with previous works in the smart home context [9,48,20] that demonstrated that users do not want to lose control over the system. P14, for example, said:

> *“I want the lights to be automatically turned off, I don’t care how to detect that I left the room .”*

Participants, however, showed to be aware of what it means being “generic.” P2, for example, pointed out that she would accept an automated solutions for simple use cases, only, e.g., to control lights and temperature, but she *“would prefer to be more in control in more complex scenarios.”* Interestingly, P24 looked at abstractions from an another point of view:

> *“there are differences between the physical world and the virtual world, e.g., the social networks. In the physical world, behaviors are simpler, and you can easily point to specific devices. In the virtual world, instead, there are many ways of doing an action, so I was more generic in such a context.”*

We verified such a statement on the collected trigger-action rules, along with the participant explanations in the final appointment. We found that, excluding one rule, all the other personalizations related to social networks generally referred to all social network platforms participants commonly used. Furthermore, while in 19 cases participants mentioned a sensor to execute a *device-centric* trigger or action, only in 12 cases participants used the *information-centric* abstraction in a very specific way, i.e., by including a specific online service such as WhatsApp. On the contrary, in 33 cases participants referred to generic notifications or warnings, without specifying any details. Despite their tendency in defining generic triggers and actions, however, participants clearly excluded fully automated solutions. P24 said:

> *"I don't want a black box. For generic triggers and actions, I would like the possibility to interact with the system, to define my preferences and eventually change the system's choices."*

Also other participants envisioned an interaction with the system with the aim of sharing their preferences and habits. Such an interaction is fundamental to *guide* the execution of generic behaviors, thus avoiding black-box solutions and maintaining a certain degree of control over the system. In explaining a rule about her private information, for example, P13 supposed to be able to define priorities over news and appointments. Instead, by explaining the rule:

> *"When I spend too much time on social networks while I'm with other people, block me all the social networks,"*

P16 said:

> *"the system could know that I'm with other people by looking at my calendar, for example."*

Knowing preferences would be fundamental for an EUD tool independently of the adopted abstraction. Let us consider the following 2 rules:

> *"When I'm hot, turn on the car car air conditioning system"* (P3);

> *"When I take a shower, I would like the shower to automatically set my ideal water temperature"* (P22).

Here, both the *people-centric* trigger of the first rule and the *device-centric* action of the second rule strongly depend on the users' concepts of *"hot"* and of *"ideal temperature"*.

5 Design Opportunities for Personalizing the IoT

In this section, we discuss the results of our diary study by identifying new design opportunities in HCI to go beyond the contemporary *vendor-centric* abstraction, with the aim of improving the relationship between the IoT, personalization paradigms, and users. We summarized our findings in 10 guidelines (Table 6). We believe that these guidelines may have a significant impact on the design of new interfaces for personalizing the IoT. By knowing the abstractions end users would adopt and in which contexts, researchers and designers may propose new solutions to break down barriers and increase EUD adoption in the field of the IoT. The guidelines, presented in the remainder of this section, have been divided in 3 categories, i.e., *adapting*, *implementing*, and *easing*. Although our work refers to trigger-action programming, the reported guidelines are generalizable to other programming paradigms.

Guideline	**Description**	**Category**
G1	TAP platforms should provide support to *teach* the programming paradigm.	Easing
G2	TAP platforms should provide support to *discover* useful and entertaining use cases to be automated.	Easing
G3	TAP platforms should allow end users to easily personalize a variety of devices and services for different use cases.	Adapting
G4	TAP platforms should provide mechanisms to adapt trigger-action rules for different subjects, e.g., for all the family.	Adapting
G5	The abstractions adopted by TAP platforms should be tailored on the part of the rule that the user is defining, i.e., the trigger or the action.	Adapting
G6	The abstractions adopted by TAP platforms should be tailored on the programming skills of the user that is defining the rule.	Adapting
G7	The abstractions adopted by TAP platforms should be tailored on the context in which the personalization is introduced.	Adapting
G8	TAP platforms should propose or allow users to select the right abstraction to be adopted.	Implementing
G9	TAP platforms should execute abstract behaviors on specific connected entities at run-time, e.g., by reasoning on users' preferences and habits and by adopting AI solutions.	Implementing
G10	TAP platforms exploiting AI should avoid black-box solutions, by providing users some mechanisms to "guide" the definition and the execution of abstract behaviors, e.g., through preference-based approaches.	

Table 6. A set of 10 guidelines extracted from the results of our study to inform the design of new EUD tools for trigger-action programming.

5.1 *Easing* the Trigger-Action Programming

As reported by by Casati et al. [10], most of the general purpose solutions for End-User Development that exist today have some limitations: they often expose too much functionality, and become too complex for non-programmers. Consequently, end users without technical skills may not find these systems useful at all: as pointed out by Bigham et al. [35], users have to deal with complex multi-layered menus to search for the right app or device to be personalized, and this can be *"confusing and intimidating."* Even in our study, during which we used a very simple notation, some participants had problems in distinguishing events and actions, while other participants had difficulties in devising innovative scenarios of IoT personalization. For these reasons, any tool for End-User Development should provide support to *teach* the adopted paradigm, e.g., trigger-action programming (**G1**). A promising direction for future works in this context could be the usage of natural language interfaces, as preliminary explored in very recent research works [40,27]: a chatbot, in particular, could "guide" the user in defining new personalizations by offering an active support in case of errors or misunderstandings, e.g., with tutorials and examples. EUD tools for trigger-action programming should also allow users to *discover* useful and entertaining use cases to be automated (**G2**). Resarchers already started to address this challenge by means of recommender systems [17,41]: recommendation techniques could make a meaningful contribution towards increasing the usability, the acceptance, and the proliferation of End-User Development [33].

5.2 *Adapting* Interfaces and Abstractions

Our study shows that end users would program a variety of devices and services in many different situations, ranging from physical environments to their virtual world, e.g., social networks and messaging services. Since the application area is very large, TAP platforms in the IoT should be tailored to users without technology experience, and should provide users with a unique place to personalize connected entities of different types for different use cases (**G3**). In many cases, in fact, participants of our study used the restrictions ***Which*** and ***Where*** to specialize a trigger-action rule for different devices, services, and locations. Furthermore, also the "social" dimension [4] should be taken into account for defining trigger-action rules (**G4**), to allow the association of rules to different subjects: some rules, e.g., *"if someone exit the room, then turn the lights off"*, could be useful if defined for the entire family, while other rules, e.g., *"if I receive an urgent mail, then notify me"*, need to be "private".

Researchers and designers in the field of end-user personalization in the IoT should also take into account that users would define their trigger-action rules by adopting different abstractions. Instead of using a single, *vendor-centric* approach, indeed, participants of our study ranged from a *device-centric* abstraction, with which they "programmed" triggers and actions of their physical entities, to other abstractions that go beyond physical devices, i.e., *information-centric* and *people-centric*. With *information-centric* triggers and actions they focused on the underlying information, while with *people-centric* triggers participants explicitly positioned themselves (and other people) *inside* the personalization. This variety of adopted abstractions highlights a huge gap with the contemporary EUD solutions, and opens the way to new design opportunities. Stemming from our findings, we claim that the abstractions adopted by TAP platforms in the IoT should be tailored on different factors. In line with related works, e.g., [52], our findings confirm that users are more specific when defining actions, thus motivating the need of adapting abstractions according to the part of the rule that is being defined (**G5**). Furthermore, we found high-level trends that suggest the need of tailoring the adopted abstractions on the programming skills of the user that is defining the rule (**G6**). Some of our participants, in particular, stated the usage of different abstractions might compensate for their lack of technological skills. Abstract *information-centric* or *people-centric* behaviors, for example, could be useful for complex use cases, where end users are not able to indicate which devices or services are needed to implement the desired behaviors. Finally, our work points to new TAP platforms that adopt different abstractions for different contexts (**G7**).

All in all, designers may explore *adaptive* interfaces for trigger-action programming by embracing different abstractions as a part of their system design space. The way a system is presented to the user, in fact, can have a priming effect on the initial mental models formed by users [13], thus influencing how they update their understanding of it based on newly-acquired knowledge [5]. Also in this case we see particular potential in empowering users to personalize their connected entities via natural language. In our recent works, for example, we developed a Conversational Search and Recommendation system that analyzes abstract user's intentions expressed in natural language and suggest pertinent IF-THEN rules that can be easily deployed in different contexts. Future works would need to further explore such an approach, e.g., by exploring how to tailor natural language dialogues according to the user's characteristics and expertise.

5.3 *Implementing* Novel Abstractions: the Role of AI and Users' Preferences

Implementing adaptive tools for trigger-action programming that take advantage of different abstractions is undoubtedly challenging for two main reasons. First, these tools should propose or allow users to select the right abstraction to be adopted (**G8**). On the one hand, novel TAP platforms could provide users with the possibility of choosing their preferred abstraction, e.g., through multi-layered interfaces. On the other hand, they could explicitly prime the user towards a specific abstraction. By reasoning on the "user profile," for example, an EUD interface could empower programming-experts in personalizing specific devices, while it could assist users with no or limited programming experience in personalizing their IoT ecosystems through *information-centric* and *people-centric* behaviors. As reported in our work, such an adaptation should also consider the context in which the personalization is introduced. To customize users' environments, e.g., homes or workplaces, designers of TAP platforms should empower users in easily personalizing physical components, be they single devices or more complex systems. For other contexts, instead, TAP platforms may be automatically adapted to different abstractions. When customizing the "online" context, for example, user interfaces could shift their abstraction towards the underlying information, while they might allow users in defining *people-centric* triggers for their health & wellbeing.

Besides selecting an abstraction, the second challenge of an adaptive tool for trigger-action programming is how to execute the intended behaviors at run-time. Indeed, while a trigger defined with the contemporary *vendor-centric* abstraction can be monitored via a specific device or online service, this is not true for *people-centric* triggers, for example, as they could be executed and adapted in different ways at run-time. The *"when I enter home"* trigger, for instance, could be monitored through a door that has been opened, or through a security camera detecting movements, among the others. The same is true for generic *information-centric* behaviors, or for those *device-centric* triggers and actions that could be associated with more than one device (e.g.,"turn on lights," that does not specify which specific lamp should be involved). An adaptive TAP platform should be therefore able to execute abstract behaviors on specific connected entities at run-time (**G9**). Also in this case, two opposite approaches could be adopted. On the one hand, new interaction techniques could be exploited to show the system choices to the users and/or ask them the final connected entities to be used. On the other hand, novel TAP platforms could automatically decide the right connected entities through which executing the intended behaviors, e.g., by reasoning on users' preferences, habits, and past interactions with the system. Future works would need to explore these different techniques, e.g., by comparing them in terms of effectiveness and users' acceptance.

Some of the approaches discussed in this section, e.g., by priming users towards an abstraction or inferring the right connected entities to be used,pave the way for the use of Artificial Intelligence (AI) solutions. While AI already has an important role in the IoT, its usage in the EUD field is still in its early stage.According to our findings, TAP platforms in the IoT should avoid black-box solutions, by providing users some mechanisms to "guide" the definition and the execution of abstract behaviors, e.g., through preference-based approaches (**G10**). Indeed, although the majority of the participants declared that they would accept an intelligent system that automatically translates abstract behaviors on real devices and services, a large number of participants stated that they would prefer to communicate with the system in order to maintain a certain degree of decision-making power. A promising approach to allow users to "control" the degree of automation of novel AI-based TAP platforms could be the integration of users' preferences into existing semantic models in the IoT field [16]. Semantic technologies can be used to infer information that has not been explicitly stated, thus facilitating the mapping between abstract information like generic users' preferences to the specific details needed to actually execute abstract triggers and actions.

6 Limitations

Our work has some limitations to be considered. The diary study has been conducted with a small sample of 24 users, all coming from the same cultural background. Although we tried to balance our participants according to different characteristics, e.g., occupations, programming experience, and tech-enthusiasm, further studies with larger and varied populations are needed to improve the generalizability of our findings. As such, our study provides a sample of typical size for qualitative studies. Also the study design suffer from some limitations and possible biases. The examples adopted in the Initial Appointment may have influenced participants. Furthermore, the choice of using trigger-action programming as the underlying notation may have biased participants and limited their expressiveness. To improve the generalizability of our findings, future works could conduct similar studies with different notations, with the aim of comparing the retrieved results. Moreover, a diary study does not allow for making any assumptions about how seriously users would engage with real-world devices, systems, and online services, that are obviously a lot harder to set up and maintain than a piece of paper. Nevertheless, using a diary empowered our participants to be creative without any restriction, and allowed us in pursuing our goal, i.e., eliciting the abstractions end users would adopt with no regard for current technological constraints. Finally, the data on which our work is based have been collected in 2017: as IoT is evolving rapidly, we must acknowledge that some of the retrieved findings may not *exactly* fit with the current IoT technological landscape. However, as the number of connected entities is continuously growing, abstraction will become increasingly important in the future: we hope that our work could serve as the basis for future EUD interfaces in this field.

7 Conclusions

Contemporary TAP platforms in the IoT present their own set of issues, and they force users to adopt a unique, *vendor-centric* abstraction that poorly adapts to the increasing complexity of the IoT. To understand whether alternative abstractions would be possible, this paper explored which abstractions end users would adopt to personalize their IoT ecosystems though a diary-based experiment with 24 participants. We showed that users would adopt different abstractions by programming devices, information, and people-related behaviors, and we demonstrated that the adopted abstraction may depended on different factors, ranging from the user profile, e.g., her programming experience, to the context in which the personalization is introduced. While users are inclined to personalize physical objects in the home, for example, they often go beyond devices in the city, where they are more interested in the underlying information. Furthermore, through *people-centric* triggers, users would explicitly position themselves (and other people) *inside* the personalization, independently of the context.

Our findings point to new design opportunities in HCI to improve the relationship between the Internet of Things, personalization paradigms, and users.

By embracing different abstractions as a part of their system design space, designers may explore EUD interfaces that go beyond the contemporary *vendor-centric* approach, able to adapt their abstractions and to share users' habits and preferences.

References

1. Akiki, P.A., Bandara, A.K., Yu, Y.: Visual simple transformations: Empowering end-users to wire internet of things objects. ACM Transaction on Computer-Human Interaction **24**(2), 10:1–10:43 (Apr 2017). https://doi.org/10.1145/3057857
2. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M.: Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys Tutorials **17**(4), 2347–2376 (Fourthquarter 2015). https://doi.org/10.1109/COMST.2015.2444095
3. Ardito, C., Buono, P., Costabile, M.F., Lanzilotti, R., Piccinno, A.: End users as co-designers of their own tools and products. Journal of Visual Languages and Computing **23**(2), 78–90 (Apr 2012). https://doi.org/10.1016/j.jvlc.2011.11.005
4. Barricelli, B.R., Valtolina, S.: End-User Development: 5th International Symposium, IS-EUD 2015, Madrid, Spain, May 26-29, 2015. Proceedings, chap. Designing for End-User Development in the Internet of Things, pp. 9–24. Springer International Publishing, Cham, Germany (2015). https://doi.org/10.1007/978-3-319-18425-8_2
5. Bibby, P.A., Payne, S.J.: Instruction and practice in learning to use a device. Cognitive Science **20**(4), 539–578 (1996). https://doi.org/10.1207/s15516709cog2004_3
6. Brackenbury, W., Deora, A., Ritchey, J., Vallee, J., He, W., Wang, G., Littman, M.L., Ur, B.: How users interpret bugs in trigger-action programming. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. pp. 552:1–552:12. CHI '19, ACM, New York, NY, USA (2019). https://doi.org/10.1145/3290605.3300782, `http://doi.acm.org/10.1145/3290605.3300782`
7. Brich, J., Walch, M., Rietzler, M., Weber, M., Schaub, F.: Exploring end user programming needs in home automation. ACM Transaction on Computer-Human Interaction **24**(2), 11:1–11:35 (Apr 2017). https://doi.org/10.1145/3057858
8. Broom, A.F., Kirby, E.R., Adams, J., Refshauge, K.M.: On illegitimacy, suffering and recognition: A diary study of women living with chronic pain. Sociology **49**(4), 712–731 (2015). https://doi.org/10.1177/0038038514551090
9. Brush, A.B., Lee, B., Mahajan, R., Agarwal, S., Saroiu, S., Dixon, C.: Home automation in the wild: Challenges and opportunities. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2115–2124. CHI '11, ACM, New York, NY, USA (2011). https://doi.org/10.1145/1978942.1979249
10. Casati, F., Daniel, F., Angeli, A.D., Imran, M., Soi, S., Wilkinson, C.R., Marchese, M.: Developing mashup tools for end-users: On the importance of the application domain. International Journal of Next-Generation Computing **3**(2) (2012)
11. Cerf, V., Senges, M.: Taking the Internet to the Next Physical Level. IEEE Computer **49**(2), 80–86 (Feb 2016). https://doi.org/10.1109/MC.2016.51
12. Ceri, S., Daniel, F., Matera, M., Facca, F.M.: Model-driven development of context-aware web applications. ACM Trans. Internet Technol. **7**(1) (Feb 2007). https://doi.org/10.1145/1189740.1189742

13. Clark, M., Newman, M.W., Dutta, P.: Devices and data and agents, oh my: How smart home abstractions prime end-user mental models. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **1**(3), 44:1–44:26 (Sep 2017). https://doi.org/10.1145/3132031
14. Corno, F., De Russis, L., Monge Roffarello, A.: A semantic web approach to simplifying trigger-action programming in the iot. Computer **50**(11), 18–24 (November 2017). https://doi.org/10.1109/MC.2017.4041355
15. Corno, F., De Russis, L., Monge Roffarello, A.: Empowering end users in debugging trigger-action rules. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. pp. 388:1–388:13. CHI '19, ACM, New York, NY, USA (2019). https://doi.org/10.1145/3290605.3300618
16. Corno, F., De Russis, L., Monge Roffarello, A.: A high-level semantic approach to End-User Development in the Internet of Things. International Journal of Human-Computer Studies **125**, 41 – 54 (2019). https://doi.org/10.1016/j.ijhcs.2018.12.008
17. Corno, F., De Russis, L., Monge Roffarello, A.: Recrules: Recommending if-then rules for end-user development. ACM Transactions on Intelligent Systems and Technology **10**(5) (Sep 2019). https://doi.org/10.1145/3344211, `https://doi.org/10.1145/3344211`
18. Corno, F., De Russis, L., Roffarello, A.M.: From users' intentions to if-then rules in the internet of things. ACM Transactions on Information Systems **39**(4) (Aug 2021). https://doi.org/10.1145/3447264, `https://doi.org/10.1145/3447264`
19. Costabile, M.F., Fogli, D., Mussio, P., Piccinno, A.: Visual interactive systems for end-user development: A model-based design methodology. IEEE Transaction on System, Man, and Cybernetics **37**(6), 1029–1046 (Nov 2007). https://doi.org/10.1109/TSMCA.2007.904776
20. Costanza, E., Fischer, J.E., Colley, J.A., Rodden, T., Ramchurn, S.D., Jennings, N.R.: Doing the laundry with agents: A field trial of a future smart energy system in the home. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 813–822. CHI '14, ACM, New York, NY, USA (2014). https://doi.org/10.1145/2556288.2557167
21. Cvijikj, I.P., Michahelles, F.: Architecting the Internet of Things, chap. The Toolkit Approach for End-user Participation in the Internet of Things, pp. 65–96. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19157-2_4
22. Czerwinski, M., Horvitz, E., Wilhite, S.: A diary study of task switching and interruptions. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 175–182. CHI '04, ACM, New York, NY, USA (2004). https://doi.org/10.1145/985692.985715
23. Danado, J., Paternò, F.: Puzzle: A mobile application development environment using a jigsaw metaphor. Journal of Visual Languages & Computing **25**(4), 297–315 (2014). https://doi.org/10.1016/j.jvlc.2014.03.005
24. Daniel, F., Matera, M.: Mashups: Concepts, Models and Architectures. Springer Publishing Company, Incorporated (2014)
25. Daniel, F., Matera, M., Pozzi, G.: Managing runtime adaptivity through active rules: The bellerofonte framework. J. Web Eng. **7**(3), 179–199 (Sep 2008)
26. De Russis, L., Corno, F.: Homerules: A tangible end-user programming interface for smart homes. In: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems. pp. 2109–2114. CHI EA '15, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2702613.2732795

27. De Russis, L., Monge Roffarello, A., Borsarelli, C.: Towards vocally-composed personalization rules in the iot. In: Proceedings of the 2nd International Workshop on Empowering People in Dealing with Internet of Things Ecosystems (EMPATHY 2021) (2021), `http://ceur-ws.org/Vol-3053/paper_1.pdf`
28. Desolda, G., Ardito, C., Matera, M.: Empowering end users to customize their smart environments: Model, composition paradigms, and domain-specific tools. ACM Transaction on Computer-Human Interaction (TOCHI) **24**(2), 12:1–12:52 (Apr 2017). https://doi.org/10.1145/3057859
29. Dey, A.K., Sohn, T., Streng, S., Kodama, J.: icap: Interactive prototyping of context-aware applications. In: Proceedings of the 4th International Conference on Pervasive Computing. pp. 254–271. PERVASIVE'06, Springer-Verlag, Berlin, Heidelberg (2006). https://doi.org/10.1007/11748625_16
30. Dixon, C., Mahajan, R., Agarwal, S., Brush, A.J., Lee, B., Saroiu, S., Bahl, P.: An operating system for the home. In: Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. pp. 25–25. NSDI'12, USENIX Association, Berkeley, CA, USA (2012)
31. Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A.G., Mehandjiev, N.: Meta-design: A manifesto for end-user development. Communications of the ACM **47**(9), 33–37 (Sep 2004). https://doi.org/10.1145/1015864.1015884
32. Ghiani, G., Manca, M., Paternò, F., Santoro, C.: Personalization of context-dependent applications through trigger-action rules. ACM Transactions on Computer-Human Interaction (TOCHI) **24**(2), 14:1–14:33 (Apr 2017). https://doi.org/10.1145/3057861
33. Haines, W., Gervasio, M., Spaulding, A., Peintner, B.: Recommendations for end-user development. In: Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI) (2010)
34. Huang, J., Cakmak, M.: Supporting mental model accuracy in trigger-action programming. In: Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 215–225. UbiComp '15, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2750858.2805830
35. Huang, T.H.K., Azaria, A., Bigham, J.P.: Instructablecrowd: Creating if-then rules via conversations with the crowd. In: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. pp. 1555–1562. CHI EA '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2851581.2892502
36. Kortuem, G., Kawsar, F., Sundramoorthy, V., Fitton, D.: Smart objects as building blocks for the internet of things. IEEE Internet Computing **14**(1), 44–51 (Jan 2010). https://doi.org/10.1109/MIC.2009.143
37. Le-Phuoc, D., Polleres, A., Hauswirth, M., Tummarello, G., Morbidoni, C.: Rapid prototyping of semantic mash-ups through semantic web pipes. In: Proceedings of the 18th International Conference on World Wide Web. pp. 581–590. WWW '09, ACM, New York, NY, USA (2009). https://doi.org/10.1145/1526709.1526788
38. Lieberman, H., Paternò, F., Klann, M., Wulf, V.: End User Development, chap. End-User Development: An Emerging Paradigm, pp. 1–8. Springer Netherlands, Dordrecht, Netherlands (2006). https://doi.org/10.1007/1-4020-5386-X_1
39. Lucci, G., Paternò, F.: Analysing how users prefer to model contextual event-action behaviours in their smartphones. In: Díaz, P., Pipek, V., Ardito, C., Jensen, C., Aedo, I., Boden, A. (eds.) End-User Development. pp. 186–191. Springer International Publishing, Cham (2015)

40. Manca, M., Parvin, P., Paternò, F., Santoro, C.: Integrating alexa in a rule-based personalization platform. In: Proceedings of the 6th EAI International Conference on Smart Objects and Technologies for Social Good. p. 108–113. GoodTechs '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3411170.3411228
41. Mattioli, A., Paternò, F.: A visual environment for end-user creation of iot customization rules with recommendation support. In: Proceedings of the International Conference on Advanced Visual Interfaces. AVI '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3399715.3399833, `https://doi.org/10.1145/3399715.3399833`
42. Munjin, D.: User Empowerment in the Internet of Things. Ph.D. thesis, Université de Genève (May 2013), `http://archive-ouverte.unige.ch/unige:28951`
43. Namoun, A., Daskalopoulou, A., Mehandjiev, N., Xun, Z.: Exploring mobile end user development: Existing use and design factors. IEEE Transactions on Software Engineering **42**(10), 960–976 (Oct 2016). https://doi.org/10.1109/TSE.2016.2532873
44. Newman, M.W.: Now we're cooking: Recipes for end-user service composition in the digital home. In: CHI 2006 Workshop IT@Home (2006)
45. Norman, D.A.: The Design of Everyday Things. Basic Books, Inc., New York, NY, USA (2002)
46. Patel, P., Pathak, A., Teixeira, T., Issarny, V.: Towards application development for the internet of things. In: Proceedings of the 8th Middleware Doctoral Symposium. pp. 5:1–5:6. MDS '11, ACM, New York, NY, USA (2011). https://doi.org/10.1145/2093190.2093195
47. Porter, M.F.: Readings in information retrieval. chap. An Algorithm for Suffix Stripping, pp. 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1997)
48. Rodden, T.A., Fischer, J.E., Pantidi, N., Bachour, K., Moran, S.: At home with agents: Exploring attitudes towards future smart energy infrastructures. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1173–1182. CHI '13, ACM, New York, NY, USA (2013). https://doi.org/10.1145/2470654.2466152
49. Sleeper, M., Melicher, W., Habib, H., Bauer, L., Cranor, L.F., Mazurek, M.L.: Sharing personal content online: Exploring channel choice and multi-channel behaviors. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. pp. 101–112. CHI '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2858036.2858170
50. Stolee, K.T., Elbaum, S.: Identification, impact, and refactoring of smells in pipe-like web mashups. IEEE Transactions on Software Engineering **39**(12), 1654–1679 (Dec 2013). https://doi.org/10.1109/TSE.2013.42
51. Truong, K.N., Huang, E.M., Abowd, G.D.: CAMP: A Magnetic Poetry Interface for End-User Programming of Capture Applications for the Home, pp. 143–160. Springer Berlin Heidelberg, Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30119-6_9
52. Ur, B., McManus, E., Pak Yong Ho, M., Littman, M.L.: Practical trigger-action programming in the smart home. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 803–812. CHI '14, ACM, New York, NY, USA (2014). https://doi.org/10.1145/2556288.2557420

53. Ur, B., Pak Yong Ho, M., Brawner, S., Lee, J., Mennicken, S., Picard, N., Schulze, D., Littman, M.L.: Trigger-action programming in the wild: An analysis of 200,000 ifttt recipes. In: Proceedings of the 34rd Annual ACM Conference on Human Factors in Computing Systems. pp. 3227–3231. CHI '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2858036.2858556
54. Zamora-Izquierdo, M.A., Santa, J., Gomez-Skarmeta, A.F.: An integral and networked home automation solution for indoor ambient intelligence. IEEE Pervasive Computing **9**(4), 66–77 (October 2010). https://doi.org/10.1109/MPRV.2010.20
55. Zaslavsky, A., Jayaraman, P.P.: Discovery in the internet of things: The internet of things (ubiquity symposium). Ubiquity **2015**(October), 2:1–2:10 (Oct 2015). https://doi.org/10.1145/2822529